Mining and Exploring Unstructured Customer Feedback Data Using Language Models and Treemap Visualizations

Cai-Nicolas Ziegler[†] Michal Skubacz[†] Maximilian Viermetz[⋆]

[†]Siemens AG, Corporate Technology, IC 1
Otto-Hahn-Ring 6, D-81730 München

[⋆]Heinrich-Heine Universität, Lehrstuhl für Informatik
Universitätsstr. 1, D-40225 Düsseldorf

{cai.ziegler, michal.skubacz}@siemens.com, viermetz@acm.org

Abstract

We propose an approach for exploring large corpora of textual customer feedback in a guided fashion, bringing order to massive amounts of unstructured information. The prototypical system we implemented allows an analyst to assess labelled clusters in a graphical fashion, based on treemaps, and perform drill-down operations to investigate the topic of interest in a more fine-grained manner. Labels are chosen by simple but effective term weighting schemes and lay the foundations for assigning feedback postings to clusters. In order to allow for drill-down operations leading to new clusters of refined information, we present an approach that contrasts foreground and background models of feedback texts when stepping into the currently selected set of feedback messages. The prototype we present is already in use at various Siemens units and has been embraced by marketing analysts.

1 Introduction

Customer feedback commonly comes in two flavors, either as structured data, such as 5-point likert scales providing ranges of valuation, or as unstructured information, that is, free-text. While structured feedback can be more easily digested by machines, mainly by applying statistical methods so as to provide an analyst with function plots, pie charts, and so forth, the degree of expressiveness allowed by structured customer forms appears largely limited. On the other hand, unstructured feedback is hard to process by machines, as the extraction of knowledge from free-text appears as a non-trivial task and much less facile to achieve than aggregating mere numbers. Still, textual feedback can provide richer information, enabling the customer to *name* issues, which is not possible for mere likert scales.

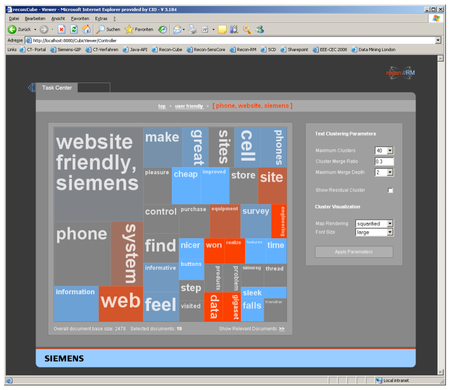
Marketing departments perform analyses of unstructured customer feedback mostly in a *manual* fashion, e.g., by having an analyst read all the comments submitted through the Web site's feedback form. This approach is bounded by the number of comments submitted, exceeding man's limited processing capabilities, and is cost-intensive. Moreover, the knowledge extracted via manual sifting through the corpus of textual comments does not necessarily reflect the actual distribution of issues: Humans are unable to process hundreds of comments in a parallel fashion, keeping track of each single issue's severity and frequency of occurrence.

We present an approach that allows an analyst to explore unstructured textual feedback in a computer-supported, graphical fashion. Our clustering-based system is geared towards the digestion of short texts, which are typical for customer comments, and employs clustering mechanics that are suited for such snippets. Salient keyphrases are selected as cluster labels, so as to provide an analyst with an overview of issues. The visual rendering of clusters is performed by means of treemap schemes.

An important aspect and contribution of our explorative clustering scheme is the contrasting of background and foreground weights, applied for determining cluster labels when performing drill-down operations.

2 Related Work

The segmentation of documents into clusters has been investigated abundantly in the past. An approach to keyphrase extraction that has become popular is the use of generative language modelling [6], as the extraction of keyphrases is important for cluster label assignment. Language models are also used to contrast two document corpora against each other, typically a domain-specific corpus against a general


recon:Cube - Viewer - Microsoft Internet Explorer provided by CIO - V 3.184
Datei Bearbeiten Ansicht Favoriten Extras ?
Zurück Suchen Favoriten
Adresse http://localhost:8080/CubicViewer/Controller
Links CT-Portal Siemens-GIP CT-Verfahren Java-API Recon-Cube Recon-SensCore Recon-RM SCD Sharepoint EEE-CEC 2008 Data Mining London
recon://RM
Task Center
top > user friendly > [phone, website, siemens]
website friendly, siemens
make
great
sites
cell
phones
pleasure
cheap
improved
store
site
control
purchase
equipment
survey
engineering
phone
system
find
nicer
won
realize
features
time
informative
buttons
products
problem
simoreg
thread
information
web
feel
step
visited
data
gigaset
sleek
falls
friendlier
Show Relevant Documents >>
Apply Parameters
SIEMENS
Local intranet

Overall document base size: 2478 Selected documents: 19

Text Clustering Parameters
Maximum Clusters: 40
Cluster Merge Ratio: 0.3
Maximum Merge Depth: 2
Show Residual Cluster: ☐

Cluster Visualization
Map Rendering: squarified
Font Size: large

Figure 1. Screenshot of our analysis platform

one, by using log-likelihood ratio tests (BLRT) [3]. Our method uses a similar scheme for contrasting two language models, applied in a cascaded fashion when drilling down into clusters.

3 System Description

The following paragraphs describe our system from a functional, user-centric view:

The analyst logs into the system and is presented a list of tasks she may execute. A task hereby represents a corpus of feedback comments, e.g., comments from an online survey as they are typically offered on Siemens Internet pages to randomly selected site visitors. Upon selecting an option offered, the analysis dashboard unfolds, see Fig. 1. The top-view gives the user an overview of frequently cited topics and issues in the form of clusters arranged in a treemap [8, 9]. Clusters are represented by rectangles, and rectangle size correlates with the number of comments assigned to one cluster. Color, as another feature dimension of clusters, may express diverse, survey-dependent aspects. In Fig. 1, colors express sentiment, i.e., the customer opinion associated with comments within each cluster (see Sec. 7). The depicted layout of clusters is inspired by ordered treemap visualizations [2]. However, our visualization method does not make use of hierarchical nesting, effectively presenting volume data in a *single*-layered fashion. Next to treemap layouts, our system also supports other means of visualizations, such as list-like renderings of clusters, vertically sorted in descending order of cluster size (see Sec. 4).

The dashboard allows the analyst to modify clustering settings, such as the number of clusters shown on the screen, various factors for cluster merging, and a flag for showing or hiding the residual cluster, which contains all comments not displayed in one of the clusters.

Moreover, the user can make the system show all the comments within the currently selected document set in a list. For the initial setup, this is the complete set of comments within the given survey, which may well reach into thousands of comments.

All rendered cluster blocks are clickable. By selecting one such rectangle, the clicked cluster becomes the new reference document set.[1] For example, by clicking on CIRCUIT BREAKERS, the set of all comments referring to circuit breakers will become the new context. The refined reference document set, reducing the initial number of comments to the number of only those contained in the selected cluster, is then clustered anew. The described operation is called "drill-down", as it allows to successively refine the set of in-focus documents.

In its current state, we allow drill-down operations to a depth of 3. The current context of clustering is hereby defined by the list of drill-downs performed so far, as shown below:

$$\textsc{Top}_{(3973)} \rightarrow \textsc{Circuit Breakers}_{(245)} \rightarrow \textsc{Low Voltage}_{(30)}$$

The subscript figures indicate the number of comments within each drill-down step, borrowed from the use case in Sec. 7. The complete survey contained 3973 comments.

As the number of documents contained in the reference document set becomes smaller with each drill-down operation, depending on the size of selected clusters and the initial comment corpus size, the analyst will eventually want the system to display the textual comments within the current context. This procedure allows the user to analyze in detail, by reading only the *relevant* comments, the issue she is particularly interested in.

4 Visualization

An important aspect of our platform's usability and benefit to users is the means of visualization we are making use of. The primordial instrument for rendering clusters is the treemap visualization technique (see, e.g., [8] and [9]), which has been proposed in the early nineties in order to visualize hierarchically organized volume data, in particular file systems and directories [8].

While treemaps are used to lay out clusters in a rectangular fashion, their usage necessitates techniques for rendering cluster labels that are able to exploit the rectangle's space as efficiently as possible.

4.1 Cluster Layouts

The applications of treemaps extend far beyond the borders of research and are nowadays implemented in all sorts

[1]We use the terms "comment" and "document" interchangeably, both denoting textual feedback from customers.

of software, such as visualizations of financial market data (*http://smartmoney.com/marketmap/*) or news text snippets on Google News (*http://marumushi.com/apps/newsmap/*). Treemaps have been designed for extremely efficient space usage, which is essential in our case. Bear in mind that all clusters are to be placed into one embracing rectangle, including cluster labels, which are expected to be legible to some point. We cannot maintain that *all* cluster labels are legible, though, as each cluster's size is defined by its share in the overall number of comments currently shown.

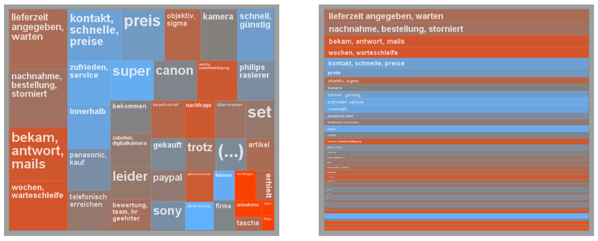
Our platform offers various types of treemap renderings, like squarified and ordered layouts [9]. However, we found that not all users were perfectly at ease with treemap visualizations. Some wished to have clusters displayed as simple lists, requiring us to include more traditional forms of visual information organization as well (see Fig. 2). We conducted a small user study, asking prospective users whether they preferred treemaps over lists or vice-versa. An overall number of 23 persons was asked upon presenting the system to them. While 6 indicated that they were more content with lists, 15 opted for treemaps. Two users were indifferent. We believe that the significant surplus of votes for treemaps derives from the fact that treemaps are much better suited for using space in an efficient manner, which is indicated by the greater average permissable font size of labels and the overall coverage of labels across the embracing view.

4.2 Labelling Clusters

Labels describe the content of clusters and consist of sequences of phrases, separated by colon (","). Each phrase, again, may consist of a sequence of words, now separated by whitespace (" "). Fig. 3 gives some examples of labels. While the assignment of cluster labels for the list-like layout is straight-forward, the assignment for treemap layouts is not so obvious: Labels have to comfortably fit into rectangles with unfavorable aspect ratios. We experimented with various strategies, like maximizing font size while breaking words whenever necessary. However, theses approaches had heavily detrimental implications on legibility and were not embraced by users. We therefore opted for another strategy, which is fairly simple but effective:

When the aspect ratio is greater than 1.3, i.e., the cluster's length is smaller than its height, we rotate the label by 90%. The font size is selected so that the longest word still fits into one single line, while not exceeding the maximum vertical space of the cluster at hand. Moreover, the algorithm tries to put as many words into one line as possible. In case our strategy either makes the label use up less than 40% of the rectangle's space or assigns a font size smaller than 8 points, we will break the longest word into two halves and start with the maximum font size over again. This heuristic is applied until the mentioned space consumption invariant is satisfied.

We found that the strategy at hand leads to smaller overall font sizes but tends to break considerably fewer words than the initial strategy. Our users were much more content with labels not filling the entire cluster rectangle in favor of preserving word integrity.



(a) (b)

Figure 2. Squarified (a) **vs. sliced** (b) **layout**

5 Data Preprocessing

Comments from customers are stored in an RDBMS. We apply common stop-word removal to all comments and put the resulting documents into an inverted index for quick access. Before index insertion, we use an external library for spell checking, as textual customer feedback tends to be particularly error-prone with respect to spelling and syntax. Stemming (e.g., classic Porter stemming [7]) is not part of our processing pipeline as we found results to become less accurate when applied.

Our platform can also be used to digest and cluster textual information from the top-n search results of major search engines, e.g., Google. Based on a given search query, extractors download either the top-n result documents or top-n result text snippets, i.e., the piece of information which search engines show in order to give an impression of the page's content. In case of the documents, another pre-processing step is necessary in order to extract the pure textual content rather than page clutter, such as advertisements, link lists, disclaimers, and so forth

To this end, we use a dedicated content extraction component [11] that removes page clutter in a fully-automated fashion.

6 Clustering and Term Weighting

At the very heart of our platform lies a document clustering algorithm. The approach we have opted for extends the work presented in [4], which performs well for short text snippets. The idea is to determine for a given reference

Rank	Cluster Labels	Sentiment
1	delivery time indicated, wait	-
2	cash on delivery, order, cancelled	-
3	contact, quick, prices	++
4	received, response, mails	--
5	content, service	++

Figure 3. Retailer's top-5 largest clusters

set D_r of documents the top-n terms that have maximum salience. Comments are then assigned to these terms whenever they contain the respective keyword. As a result, we obtain n clusters, plus a residual cluster that contains all comments $d \in D_r$ which have not been assigned to one of the n clusters. Clusters do not have to be disjoint, i.e., overlaps are allowed.

6.1 Computing Term Weights

The weight assignment scheme is based on tf·idf scores [1]. Suppose that t_i is a term that appears in some comment d in reference set D_r. Function $\mathrm{df}(t_i)$ denotes the document frequency, that is, the number of documents in D_r that contain t_i. Next, function $\mathrm{tf}^+(t_i)$ computes the summed term frequency for t_i:

$$\mathrm{tf}^+(t_i) = \sum_{d \in D_r} \mathrm{tf}(t_i, d), \tag{1}$$

where $\mathrm{tf}(t_i, d)$ is the term frequency of t_i in document d, i.e., the number of times that t_i occurred in d. The eventual weight $w_r(i)$ of term t_i with respect to reference set D_r is computed as follows:

$$w_r(i) = 1 + \log(\mathrm{tf}^+(t_i)) \cdot \log(\frac{|D_r|}{\mathrm{df}(t_i)}) \tag{2}$$

Next to the above shown term weighting scheme we also implemented a language modelling approach to keyphrase weighting (see, e.g., [5]), where the salience of terms for a reference document set is determined by matching against term frequency distributions of a generalized language model. Top-weighted terms are those that are *typical* for the reference set and *atypical* for the general model [3].

We are planning to set up an in-vivo user study with a control group of users to evaluate which of the two models gives better accuracy in selecting salient terms.

6.2 Foreground vs. Background Weights

In order to optimize the process of hierarchical exploration of the entire set of textual comments, we propose a method that contrasts foreground and background weights for terms, in order to enable context-awareness:

The *foreground* D_f is the reference document set of the current context. For instance, after drilling from the top-level into cluster c, labelled CIRCUIT BREAKERS, the new foreground is defined by all comments contained in c, i.e., $D_f := c$. The *background* D_b for the case at hand would be all documents excluded from the current context, i.e., $D_b := D \setminus D_f$, where D denotes the complete set of comments.

For each term t_i, we compute its foreground weight $w_f(i)$ and its background weight $w_b(i)$, where D_f, respectively D_b, are used as reference sets for the weight function. These two weights then serve as means to compute the final weight $w(i)$ for term t_i in the current clustering step:

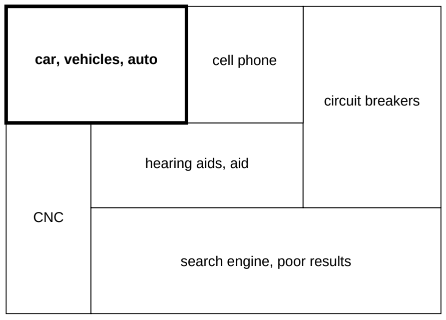
$$w(i) = \frac{w_f(i)}{w_b(i)} \cdot \log(w_f(i) + w_b(i)) \tag{3}$$

The rationale behind this approach is illustrated by the example given in Fig. 4: At first, there is no context selected, the whole document set serves as reference, as is the case in Fig. 4(a). Topics are widely spread and exhibit little semantic coherence. Then the analyst drills into cluster c, CARS, VEHICLES, AUTO, which becomes her new context. In order to support the user in her endeavor to *focus* on the selected topic, the new clustering of documents in c needs to be context-aware: The new cluster labels should be *typical* for the context, i.e., foreground c, and *atypical* for the background. Hence, we take the ratio of foreground and background weight, so as to boost terms with high foreground and low background weight, enforcing the context. For instance, suppose there was a term, t_x, whose weight is high both for the complete set as well as the current context. Even though it is an overall highly salient term, it is *not typical* for the context, so we want to intuitively penalize t_x. The effect can be seen in Fig. 4(b), which is very much focused on automobiles and car-related topics.

The ratio of foreground and background weights is multiplied with the log of sums of $w_f(i)$ and $w_b(i)$. The rationale behind taking the sum is to encourage terms that have high foreground *and* high background weights, as opposed to terms t_i that have both low $w_f(i)$ and $w_b(i)$: When only taking the ratio of both weights, it could be the case that a term with low $w_f(i)$ and even lower $w_b(i)$ gets a large overall weight, because of a high *ratio*. The log is used for dampening.

6.3 Merging Clusters

Upon sorting term weights $w(i), i \in \{1, 2, \ldots, |D_r|\}$ in descending order, we perform one last step before generating clusters: As mentioned in Sec. 6, clusters may overlap in the documents they contain. If overlap between clusters c_k



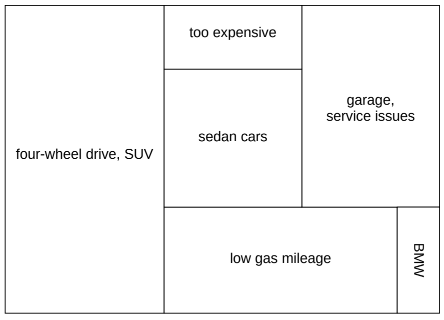


Figure 4. Top-level clustering (a) and clustering for context CAR, VEHICLES, AUTO (b)

and c_l exceeds a configurable threshold[2], they are merged.

Next, a bigram check is performed. That is, if the two labels are colocated within a sufficiently large share of documents, typically around 50%, and if the order of colocation is fixed, e.g., BREAKERS always follows CIRCUIT, then the merged labels are regarded as a *bigram*, as is the case for CIRCUIT BREAKERS. If the bigram check fails, the label strings for c_k and c_l become separated by a semicolon, ",", as shown in Fig. 4.

The new list of labels can again be analyzed for merging. The number of loops is bounded by a user-configurable parameter. Finally, the top-n labels are chosen for clusters.

7 Usage Scenario

The system has been applied to numerous customer feedback corpora within Siemens, as well as to non-corporate feedback corpora for training and performance tuning. The following paragraphs describe typical analysis setups and also give some evidence on how fast the analyst may converge when drilling from our tool's top layer down to specific information pieces she is interested in.

While the platform has been designed with explicit customer feedback in mind, it may also be applied to other problems, such as the clustering of search engine results for a given search query (see Sec. 5). However, note that the clustering techniques have been optimized for shorter text snippets rather than full-blown documents.

7.1 Analysis Use Cases

The main scenario for our platform is online surveys offered on Siemens Internet sites across all business units. These surveys assess the visitor's satisfaction with Siemens as well as the website at hand. Two free-text questions are contained, namely *"Why did you not achieve your goal?"*, which was asked when the user had indicated discontent, and *"What is your overall comment?"*. For example, in case of one Siemens business unit, there were 3973 answers for the first and 2754 answers for the second question. The surveys are largely focused on *structured* feedback, offering around 40 questions with ratings on 5-point likert scales. One of these likert-scale questions asks the user to specify her level of satisfaction. We use this machine-readable information to enrich our cluster visualization with sentiment information: As shown in Fig. 1, colors range from red to blue. The first expresses utter discontent, the latter full satisfaction. Sentiment scores for one given cluster are averaged across contained comments.

The use of polarity information allows the analyst to focus on the topics that are "hot", i.e., bright red, and are perceived as problems by customers. For instance, we found that US customers are dissatisfied with one product line, as Siemens had cut down on support services during the last two years for this particular product line.

Siemens customer feedback corpora other than the described breed of online surveys in our Internet portals are comprised of data containing transcripts of customer calls, mails from customers sent to the contact address, and related forms of textual feedback.

Fig. 3 gives an example of real-life clusters as they have been identified by our engine. The shown clusters have not been computed based on Siemens corpora, but on consumer feedback referring to a German retail shop for electronics equipment. We garnered the respective comments from an online forum containing ratings of electronics retailers.[3]

7.2 Drill-Down Speed

An interesting aspect is the drill-down convergence rate, when moving from the top layer, i.e., the state where the

[2]Typical threshold values are within the range $[0.4, 0.6]$.

[3]We translated the labels from German to English for ease of reading.

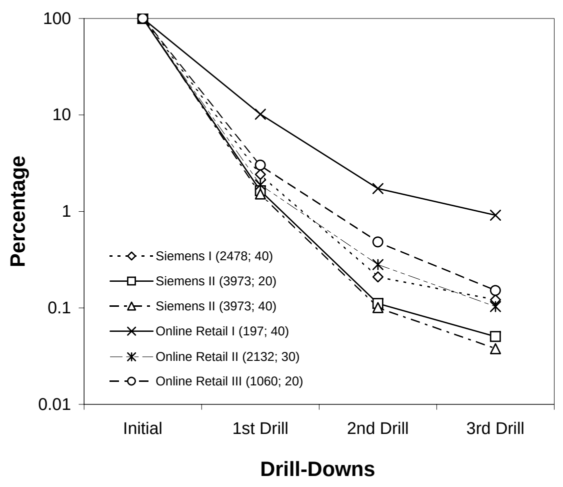


Figure 5. Drill-downs allow to focus quickly

current focus comprises all comments, to the bottom layer, which is reached upon performing three consecutive drill-down operations. To this end, Fig. 5 gives the share of in-focus comments with regard to the initial comment corpus size. For each drill-down step, the average share is computed by taking the average number of comments per currently displayed clusters. We tested various customer feedback corpora, with different settings such as the number of clusters displayed in one screen and the initial size of the comment corpus. For each trend line in Fig. 5, these parameters are indicated in brackets.

The study at hand confirms that the analyst is guided very quickly from the top perspective to the specific piece of information she is interested in, leaving on average no more than 2.2 comments in focus after only three drill-downs, virtually irrespective of the initial corpus size. The first drill-down reduces the focus set to an average of 4.2% of the initial corpus, the second goes down to as low as 0.9%.

We are well aware that further empirical analyses are required to corroborate our hypothesis that the speed of focusing on issues does not mean that important topics become discarded or dropped.

8 Conclusion and Outlook

We have presented our platform for mining and exploring massive unstructured customer feedback. The system is geared towards real-world usage and maintains its focus on practicality and usability.

From a technical point of view, our major contribution is the concept of contrasting foreground and background weights of prospective cluster labels when drilling down into the document set. Users have greatly embraced the quick focusing of automated clustering on the given context. However, we have no empirical analyses in order to underline our approach's better performance with respect to traditional approaches that use one model only. These evaluations are hard to perform as there is no such thing as a labelled gold-standard dataset that can be used to analyze clustering quality. On-site feedback of user experience appears as the only option, necessitating large numbers of users to compare diverse clustering results. This is one of the aspects we would like to work on in the future.

From a more practical point of view, we are currently integrating the customer feedback tool into the larger context of our reputation monitoring platform [10].

References

[1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, Reading, MA, USA, May 1999.

[2] B. Bederson, B. Shneiderman, and M. Wattenberg. Ordered and quantum treemaps: Making eeffective use of 2D space to display hierarchies. *ACM Transactions on Graphics*, 21(4):833–854, 2002.

[3] T. Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74, 1993.

[4] M. Gamon, A. Aue, S. Corston-Oliver, and E. Ringger. Pulse: Mining customer opinions from free text. In *Proceedings of the 6th Symposium on Intelligent Data Analysis*, volume 3646 of *LNCS*, pages 121–132, Madrid, Spain, 2005. Springer-Verlag.

[5] G. Mishne and M. de Rijke. Language model mixtures for contextual ad placement in personal blogs. In *Proceedings of the FinTAL Conference*, volume 4139 of *LNCS*, pages 435–446, Turku, Finland, 2006. Springer-Verlag.

[6] J. Ponte and B. Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 275–281, Melbourne, Australia, 1998. ACM Press.

[7] M. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[8] B. Shneiderman. Tree visualization with tree-maps: 2-d space-filling approach. *ACM Transactions on Graphics*, 11(1):92–99, 1992.

[9] B. Shneiderman and M. Wattenberg. Ordered treemap layouts. In *IEEE Symposium on Information Visualization*, pages 73–78, San Diego, CA, USA, 2001. IEEE Press.

[10] C.-N. Ziegler and M. Skubacz. Towards automated reputation and brand monitoring on the web. In *Proceedings of the IEEE International Conference on Web Intelligence*, pages 1066–1070, Hong Kong, China, December 2006. IEEE Computer Society Press.

[11] C.-N. Ziegler and M. Skubacz. Content extraction from news pages using particle swarm optimization on linguistic and structural features. In *Proceedings of the IEEE International Conference on Web Intelligence*, pages 242–249, Fremont, CA, USA, November 2007. IEEE Computer Society Press.